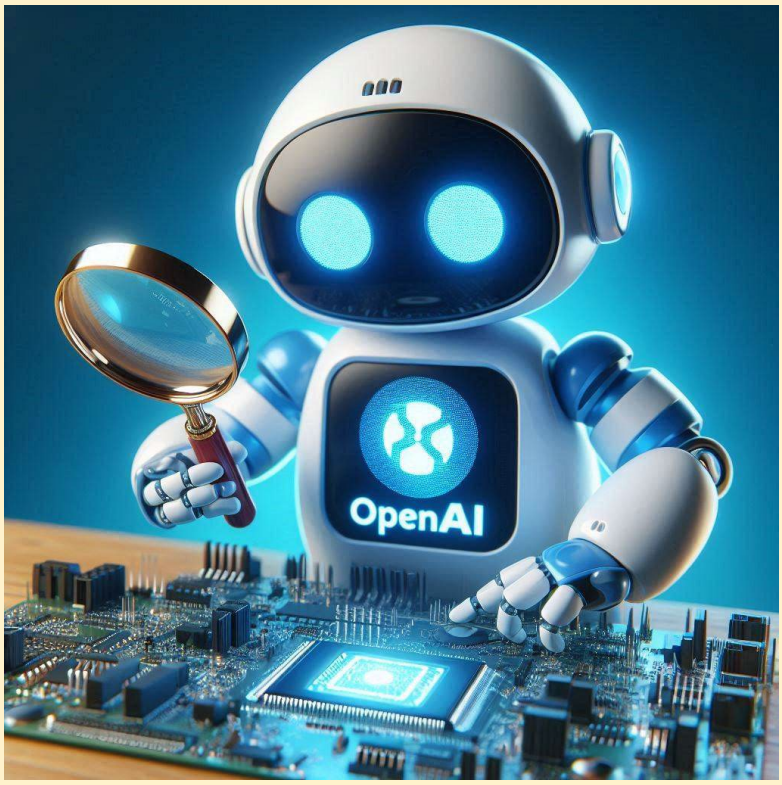
TECHWIZ 5
A GLOBAL I.T. COMPETITION

Aptech
Unleash your potential



Software Requirements Specification
Version 1.0

ApGENie

Theme: Question Bot

Project Name: ApGENie

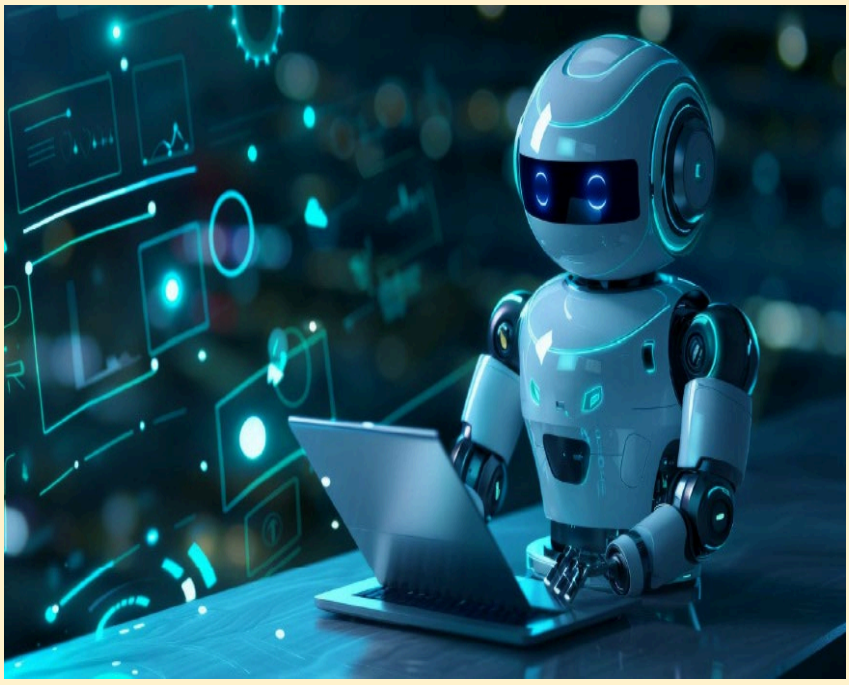
Category: GenAI Smart Solutions

Contents

1.1 Background and Necessity for the Application

Financial services industry is continually evolving, driven by advancements in technology and increasing consumer expectations for faster, more accurate, and personalized services. Traditional methods of handling and processing loan-related queries, such as those for housing loans, education loans, vehicle loans, and medical loans are often labor-intensive and time-consuming. These methods can lead to delays, inaccuracies, and a lack of personalized customer service, which in turn can result in customer dissatisfaction and operational inefficiencies.

In this context, the use of Generative AI (Gen AI) and Large Language Models (LLMs) such as those provided by OpenAI present a transformative opportunity. Gen AI and LLMs can automate and streamline the processing of complex and varied customer queries. By leveraging these advanced AI technologies, the proposed application can offer accurate responses to user queries, thereby enhancing the user experience and operational efficiency. The integration of Gen AI and LLMs ensures that the application can understand and respond to user queries providing comprehensive and relevant information tailored to each user's specific requirements. This innovation not only addresses the current limitations in handling large volumes of diverse loan-related queries, but also sets a new standard for customer service in the financial sector.



1.2 Proposed Solution

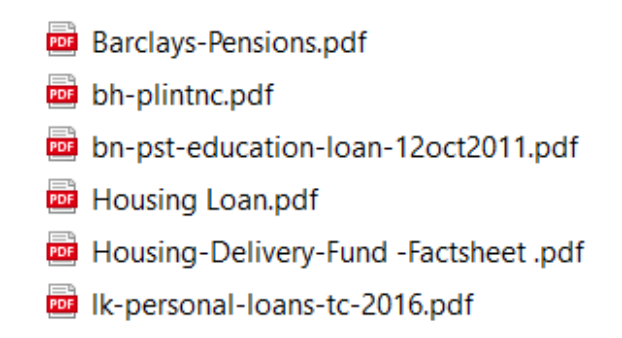
'**ApGENie**' aims to develop an intelligent application that efficiently handles user queries about various loan types using GenAI, LLM, and OpenAI technologies. This application will accept PDF documents as inputs from categories such as housing loans, education loans, vehicle loans, and medical loans. The PDF files for these categories are provided for this project. The application will use Optical Character Recognition (OCR) and Natural Language Processing (NLP) to extract and digitize relevant text and data from PDFs for further processing. The extracted data will be preprocessed to clean and remove noise, irrelevant information, and inconsistencies ensuring high-quality and accurate data for subsequent stages.

Key features and variables relevant to each loan type will be extracted from the preprocessed data, forming the basis for understanding and processing user queries. The text data will be converted into embeddings using pre-trained models (OpenAI LLMs such as ChatGPT), providing a meaningful representation of the data and enabling the application to understand better and process the information.

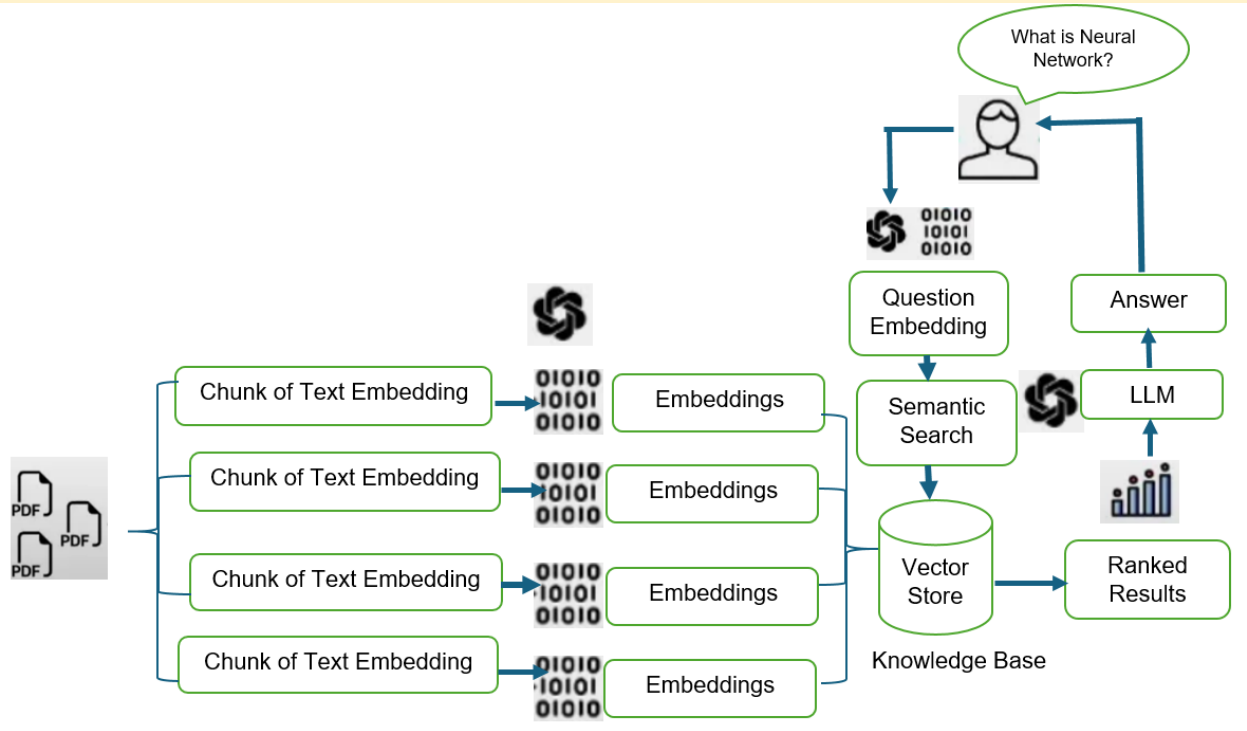
User queries will be processed and matched with the extracted features from the data using chunking strategies, re-rankers, and generation prompts to effectively address the queries. An advanced search system, powered by Gen AI and LLMs, will leverage the processed data and embedding models to provide precise responses. An interface will be integrated to interact with users, using the processed data and model outputs to answer queries in real-time, ensuring a seamless user experience.

The final responses, search results, or generated text outputs will be presented to the users, ensuring they receive accurate and relevant information based on their queries.

Hint: Here is a sample of the PDFs from the Finance domain category provided for implementation purposes:



The architecture for building the application will be as follows:



Sample Architecture of the Application

Key architectural elements of the application are as follows:

1. **User Interface (UI):** The chatbot provides a seamless platform for users to input questions and receive answers including handling PDF queries from the finance domain.

2. **Natural Language Understanding (NLU):** The NLU processes user queries, interpreting those using NLP techniques such as tokenization, Named Entity Recognition (NER), and parsing.
3. **Vector Store:** This component efficiently retrieves information by storing vector representations of text chunks from PDFs, created using language models.
4. **Embeddings:** Embeddings capture semantic and contextual information, converting text into dense vector representations for effective retrieval.
5. **LLMs:** LLMs enhance the chatbot's language understanding and generation capabilities, allowing interactive and context-aware dialogue.
6. **Conversational Retrieval:** The chatbot retrieves relevant information by comparing user query vectors with those in the Vector Store, identifying the most pertinent text.
7. **Chat History and Context:** The chatbot tracks conversation history to provide coherent and contextually relevant responses.

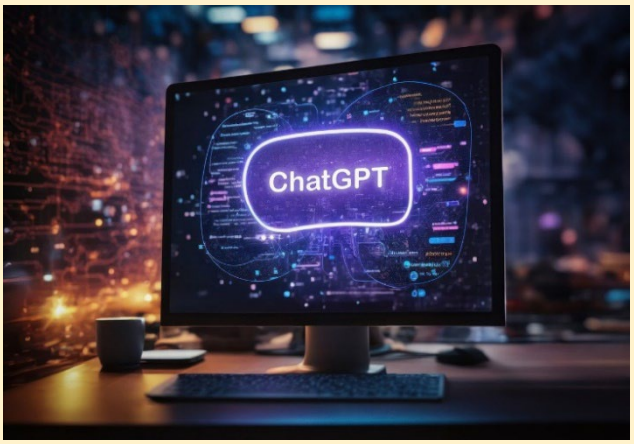
The development phase of the application includes following steps:

1. Input (PDF Documents)
2. OCR and Text Extraction
3. Data Preprocessing
4. Feature Extraction
5. Text Embeddings
6. User Query Processing
7. Search System and Chatbot
8. Output (Responses to User Queries)

1.3 Purpose of the Document

The purpose of this document is to present a detailed description of the Gen AI simulating human conversation by matching user prompts to scripted responses titled '**ApGENie**'.

This document explains the purpose and features of Gen AI and LLM simulation of human conversation and the constraints under which it should operate. This document is intended for both stakeholders and developers of the application.



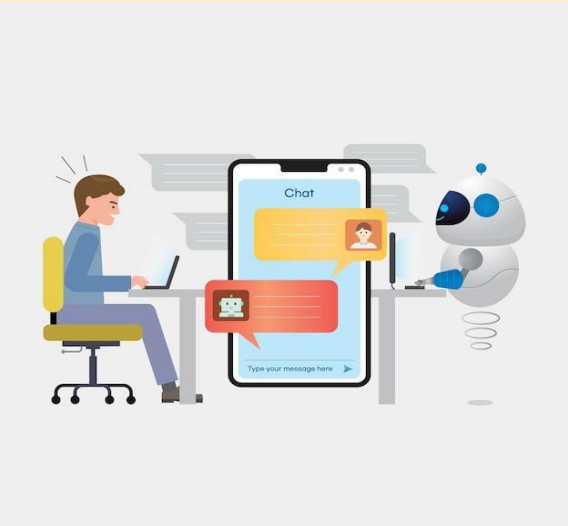
1.4 Scope of Project

The '**ApGENie**' application should be able to assist users process PDF documents related to various loan types such as housing loans, education loans, vehicle loans, and medical loans.

Key features will include OCR and NLP for text extraction, data preprocessing, and conversion of text data into embeddings using pre-trained models. Advanced search techniques and an interface will be employed to provide accurate and relevant real-time responses to user queries, ensuring a seamless user experience.

1.5 Constraints

Constraints for the '**ApGENie**' project could include the requirement for robust handling of varying document formats and layouts, which may impact the accuracy of OCR. Ensuring consistent data extraction is essential amidst these challenges. Additionally, Security and Privacy concerns when handling sensitive loan-related information will also be critical, requiring adherence to data protection regulations and implementing secure data handling practices throughout the application's lifecycle.



1.6 Functional Requirements

The project involves developing an intelligent question-and-answer application utilizing Gen AI, LLMs, and technologies from OpenAI. The application will process user input from PDF documents, enabling it to handle various structured and unstructured data formats commonly found in financial documents. Some of the functional requirements are explained as follows:

i. **PDF Input Handling** - Users will upload PDF documents containing queries or information related to different loan categories. The application will employ OCR to extract text data from the PDFs provided.

ii. **Text Preprocessing** - The extracted text will undergo preprocessing to clean and structure the data making it suitable for analysis. This includes tasks such as tokenization, normalization, and removal of irrelevant content.

iii. **Chunking Strategies** - The preprocessed text will be segmented into meaningful chunks to facilitate better understanding and processing by the AI models.

iv. **Embedding Models** - The segmented text chunks will be converted into dense vector representations using advanced embedding models. These embeddings capture the semantic meaning of the text enabling effective comparison and retrieval.

v. **Query Processing** - When a user submits a query, the application will process it using Gen AI and LLMs to understand the intent and context. The query will be matched against the relevant text chunks to find the most pertinent information.

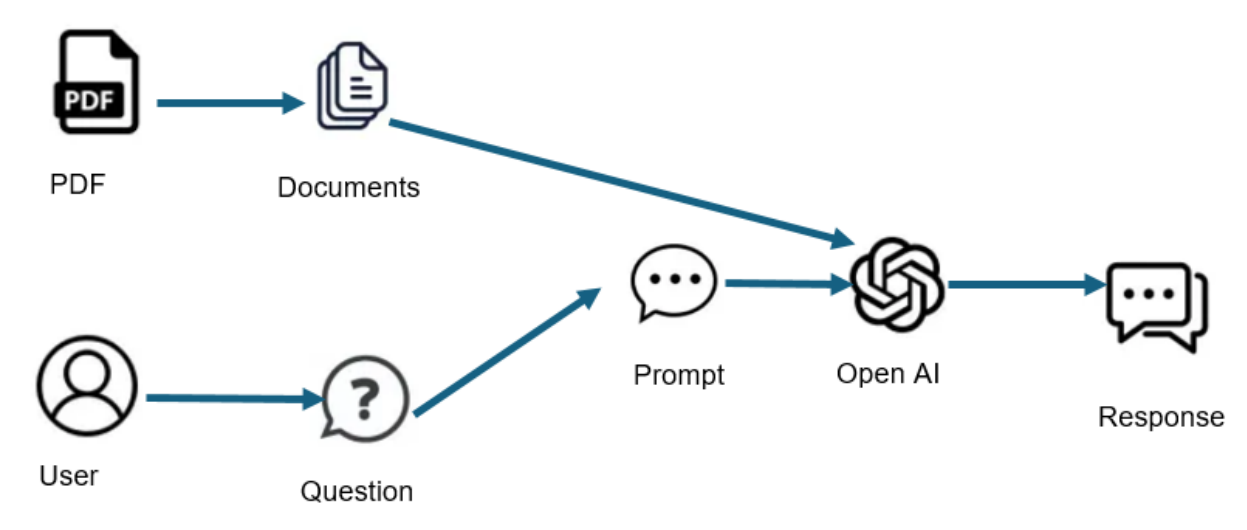
vi. **Re-Ranking** - The retrieved text chunks will be re-ranked based on their relevance to the query, ensuring that the most accurate and useful information is presented to the user.

vii. **Response Generation** - Using Gen AI, the application will generate a coherent and contextually appropriate response to the user's query. This response will be tailored to the specific requirements of the user, providing comprehensive and precise information.

viii. **User Interface** - The response will be delivered to the user through an intuitive Web interface, allowing for easy interaction and follow-up queries if required.

The '**ApGENie**' will receive input in the form of PDF documents and users will provide queries. It will scan these documents and respond by extracting information directly from the PDFs.

'**ApGENie's**' workflow will be as follows:



Following is a sample of an expected output:

```
[73]  1 # Read the user query
      2
      3 query = input()
```

What is the benefit for physically handicapped children in case of change in member's family status?

Prompt Entered by User

Search Layer Output:

```
1 top_3_RAG
```

	Documents	Metadatas
1	(1) marriage or establishment of a Civil Union...	{'page': 32, 'text': '(1) marriage or establis...
4	(1) the child is incapable of self-support as ...	{'page': 39, 'text': '(1) the child is incapab...
3	Section A – Eligibility\nMember Life Insurance...	{'page': 7, 'text': 'Section A – Eligibility M...

Search Layer Output

Generation Layer Output:

```
[92]  1 # Print the response
      2
      3 print("\n".join(response))
```

In the case of a change in the member's family status, the benefit for physically handicapped children would depend on the specific insurance policy. To provide an accurate answer, I r

After analyzing the relevant documents, I found the following information:

1. Document: Policy Name (Page Number)
   - Policy 1: (1) marriage or establishment of a Civil Union (Page 32)
   - Policy 2: (1) the child is incapable of self-support as a result of severe mental retardation, cerebral palsy, epilepsy, autism, or some other physical or mental incapacity (Page
   - Policy 3: Section A – Eligibility (Page 7)

Based on the available information, Policy 1 mentions the possibility of a benefit in case of a change in the member's family status, but it does not specifically mention physically h

Considering the above information, the policy documentation does not explicitly mention the benefit for physically handicapped children in case of a change in family status. It is recc

Please note that the provided answer is based on the available documents and may not cover all possible scenarios. It is always a good practice to refer to the policy documents for acc

Citations:

- Policy 1: Page 32, Policy Name
- Policy 2: Page 39, Policy Name
- Policy 3: Page 7, Policy Name

```
1 top_3_RAG
```

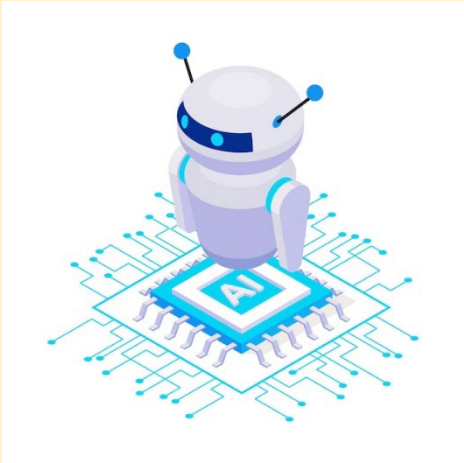
	Documents	Metadatas
1	(1) marriage or establishment of a Civil Union...	{'page': 32, 'text': '(1) marriage or establis...
4	(1) the child is incapable of self-support as ...	{'page': 39, 'text': '(1) the child is incapab...
3	Section A – Eligibility\nMember Life Insurance...	{'page': 7, 'text': 'Section A – Eligibility M...

1.7 Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the application. These are listed as follows:

1. **Scalable**: The application must handle increasing numbers of users and documents without significant degradation in performance.
2. **Secure**: The application should be designed to prevent unauthorized users from gaining access to it.
3. **Accuracy**: OCR and NLP processes should achieve high accuracy rates in text extraction and understanding loan-related content.

4. **Performance**: The application should respond to user queries within seconds, even with large PDF documents.
5. **Usable**: The UI should be intuitive and accessible, facilitating easy interaction for users of varying technical backgrounds.



These are the bare minimum expectations from the project. It is a must to implement the FUNCTIONAL and NON-FUNCTIONAL requirements given in this SRS.

Once they are complete, you can use your own creativity and imagination to add more features if required.

1.8 Interface Requirements

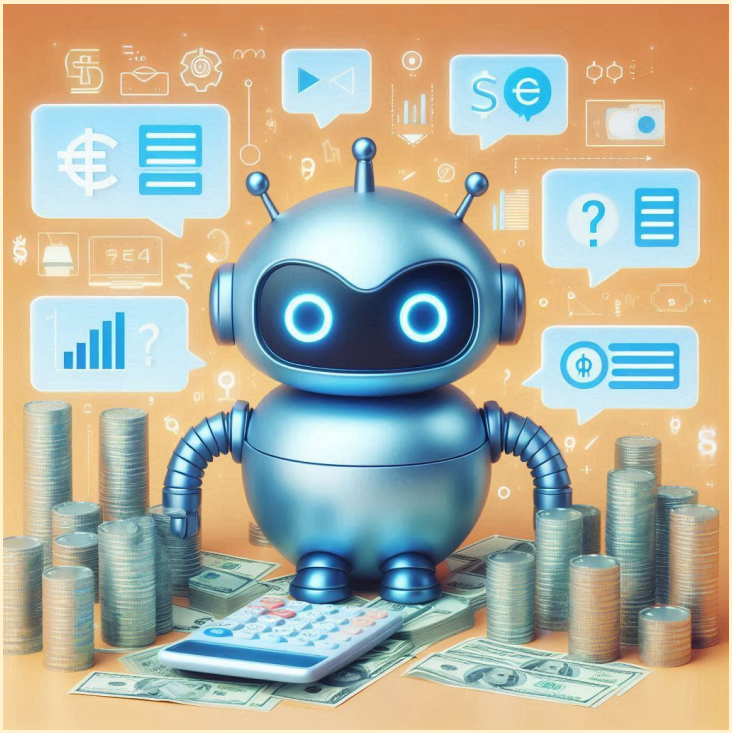
1.8.1 Hardware

Intel Core i5/i7 Processor or higher
8 GB RAM or higher
Color SVGA
500 GB Hard Disk space
Keyboard and Mouse

1.8.2 Software

Technologies to be used:

1. **Frontend**: HTML5 or any other scripting langauges
2. **Backend**: Flask/Django
3. **Data Store**: PDF
4. **Programming/IDE**: Python, Jupyter Notebook, Anaconda, or Google Colab
5. **Libraries**: Tensorflow, NLTK, Keras, OpenAI API, Python libraries, and pre-trained transformers



1.9 Project Deliverables

You will design and build the project and submit it along with a complete project report that includes:

- Problem Definition
- Design Specifications
- Diagrams such as User Flow Diagram/User Journey Map
- Detailed steps to execute the project
- Test Data Used in the Project
- Project Installation Instructions
- Link of GitHub for accessing the uploaded project code (Link should have public access)
- Link of Published Blog

The source code, including .ipynb files for Jupyter Notebook and Google Colab, should be shared via GitHub. Appropriate access permissions should be granted to users to allow testing for Jupyter Notebook and Google Colab. The consolidated project must be submitted on GitHub with a ReadMe.doc file listing assumptions (if any) made at your end.

Provide the GitHub URL where the project has been uploaded for sharing. The repository on GitHub should have public access. Documentation is a very important part of the project; hence, all crucial aspects of the project must be documented properly. Ensure that documentation is complete and comprehensive.

You should publish a blog of minimum 2000 words on any free blogging Website such as Blogger, Tumblr, Ghost or any other blogging Website. The link of the published blog should be submitted along with the project documentation.

Submit a video (.mp4 file) demonstrating the working of the application, including all the functionalities of the project. This is MANDATORY.

~~~ End of Document ~~~